



US 20090006488A1

(19) **United States**

(12) **Patent Application Publication**  
**Lindahl et al.**

(10) **Pub. No.: US 2009/0006488 A1**

(43) **Pub. Date: Jan. 1, 2009**

(54) **USING TIME-STAMPED EVENT ENTRIES TO  
FACILITATE SYNCHRONIZING DATA  
STREAMS**

**Publication Classification**

(51) **Int. Cl.**  
**G06F 17/30** (2006.01)

(52) **U.S. Cl.** ..... **707/201; 707/E17.001**

(57) **ABSTRACT**

An embodiment of a system to synchronize data streams is described. During operation, the system receives a first data stream having a first data type and performs a sequence of operations on at least a portion of the first data stream. Next, the system stores metadata associated with the first data stream one or more times during the sequence of operations. Metadata can include a time stamp, data-stream information associated with the portion of the first data stream, and event information associated with the occurrence of one or more events during the sequence of operations. Moreover, the system provides a notification based on the stored metadata, the notification to facilitate synchronizing the first data stream with a second data stream having a second data type.

(76) Inventors: **Aram Lindahl**, Menlo Park, CA (US); **Anthony J. Guetta**, Palo Alto, CA (US); **Joseph M. Williams**, Dallas, TX (US); **Dan Timis**, Mountain View, CA (US)

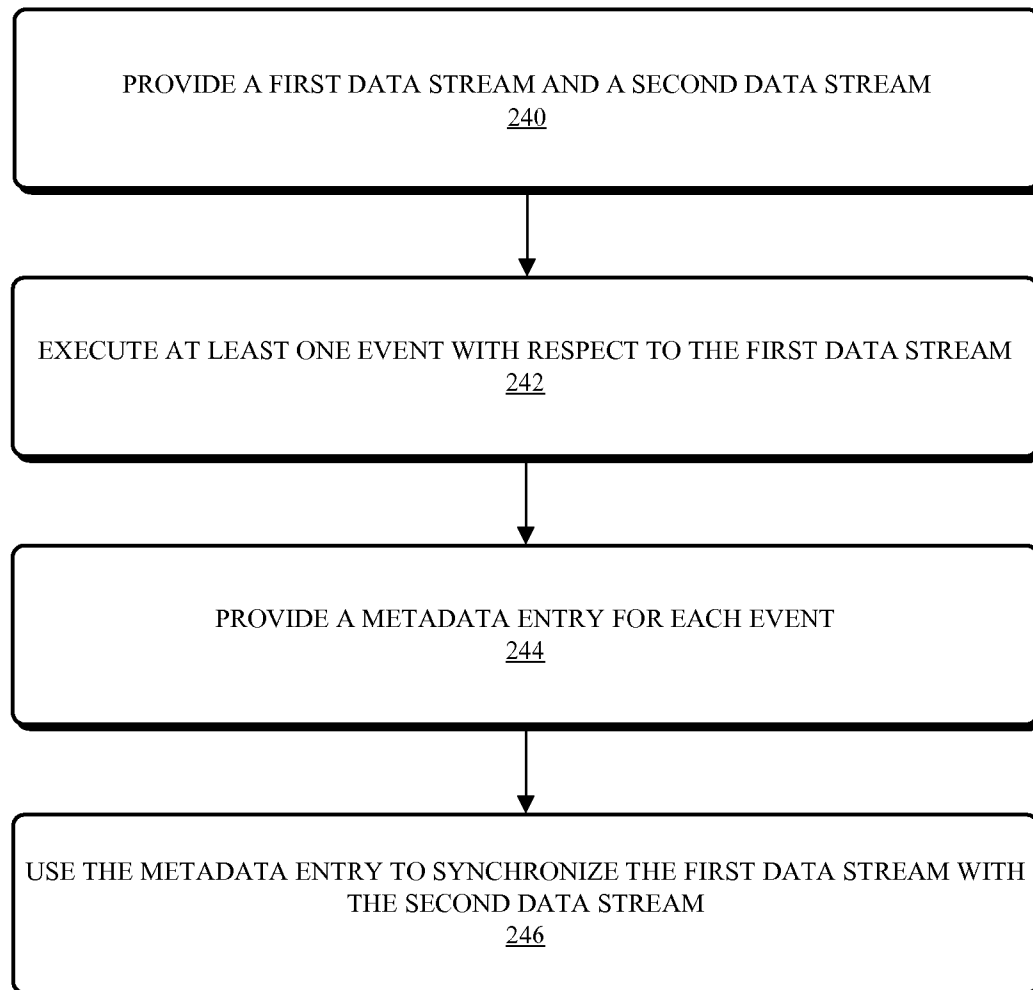
Correspondence Address:

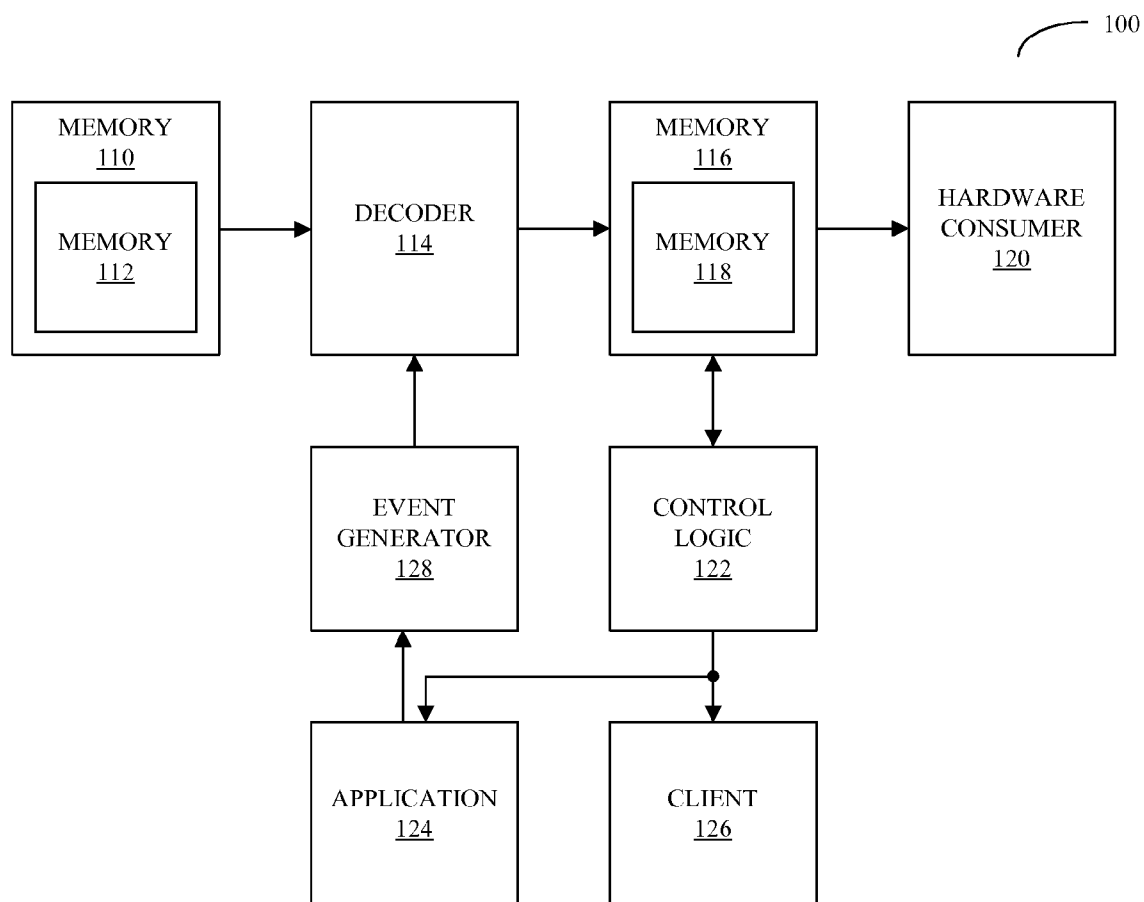
**PVF – APPLE INC.**  
**c/o PARK, VAUGHAN & FLEMING LLP**  
**2820 FIFTH STREET**  
**DAVIS, CA 95618-7759 (US)**

(21) Appl. No.: **11/770,507**

(22) Filed: **Jun. 28, 2007**

— 230



**FIG. 1**

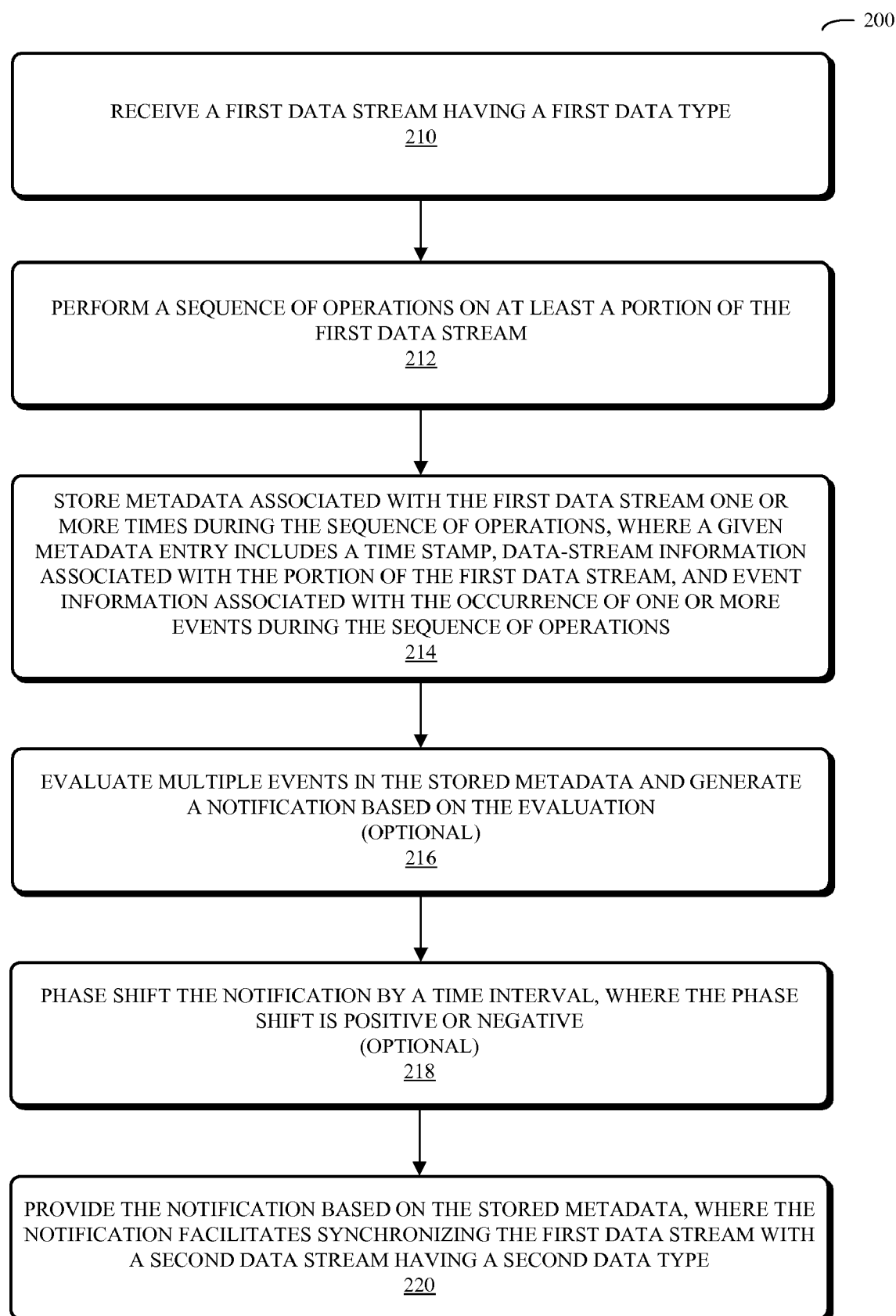
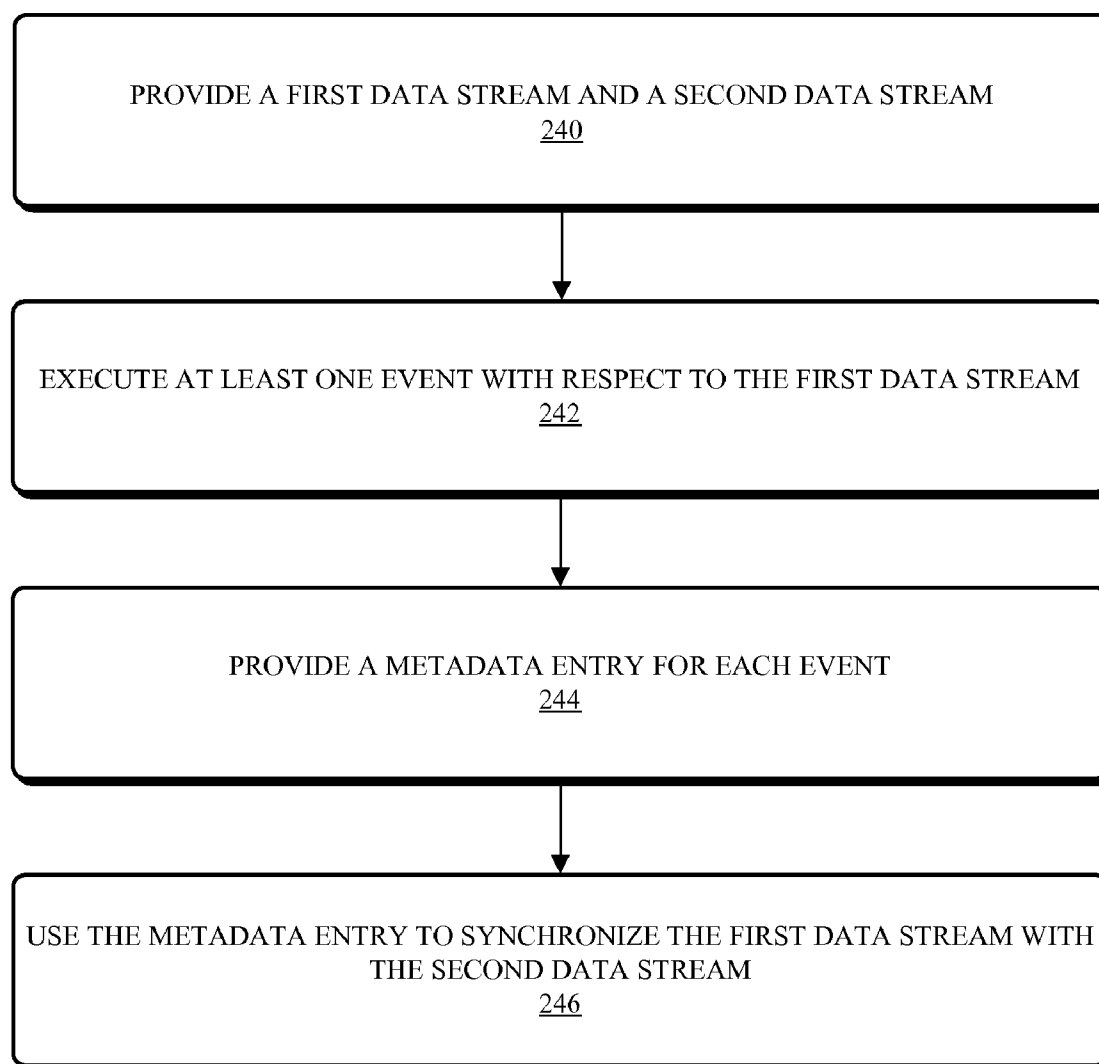


FIG. 2A

230

**FIG. 2B**

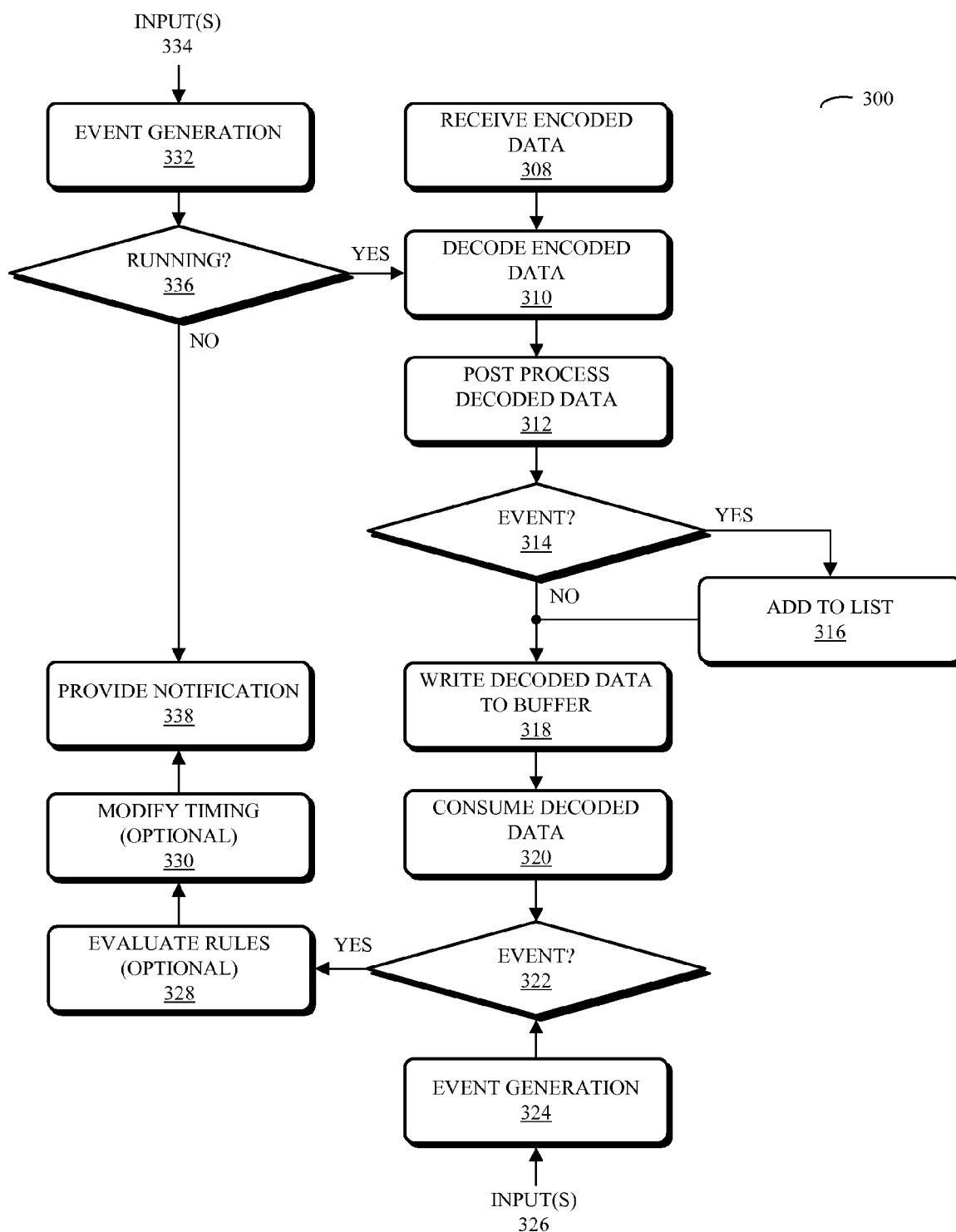
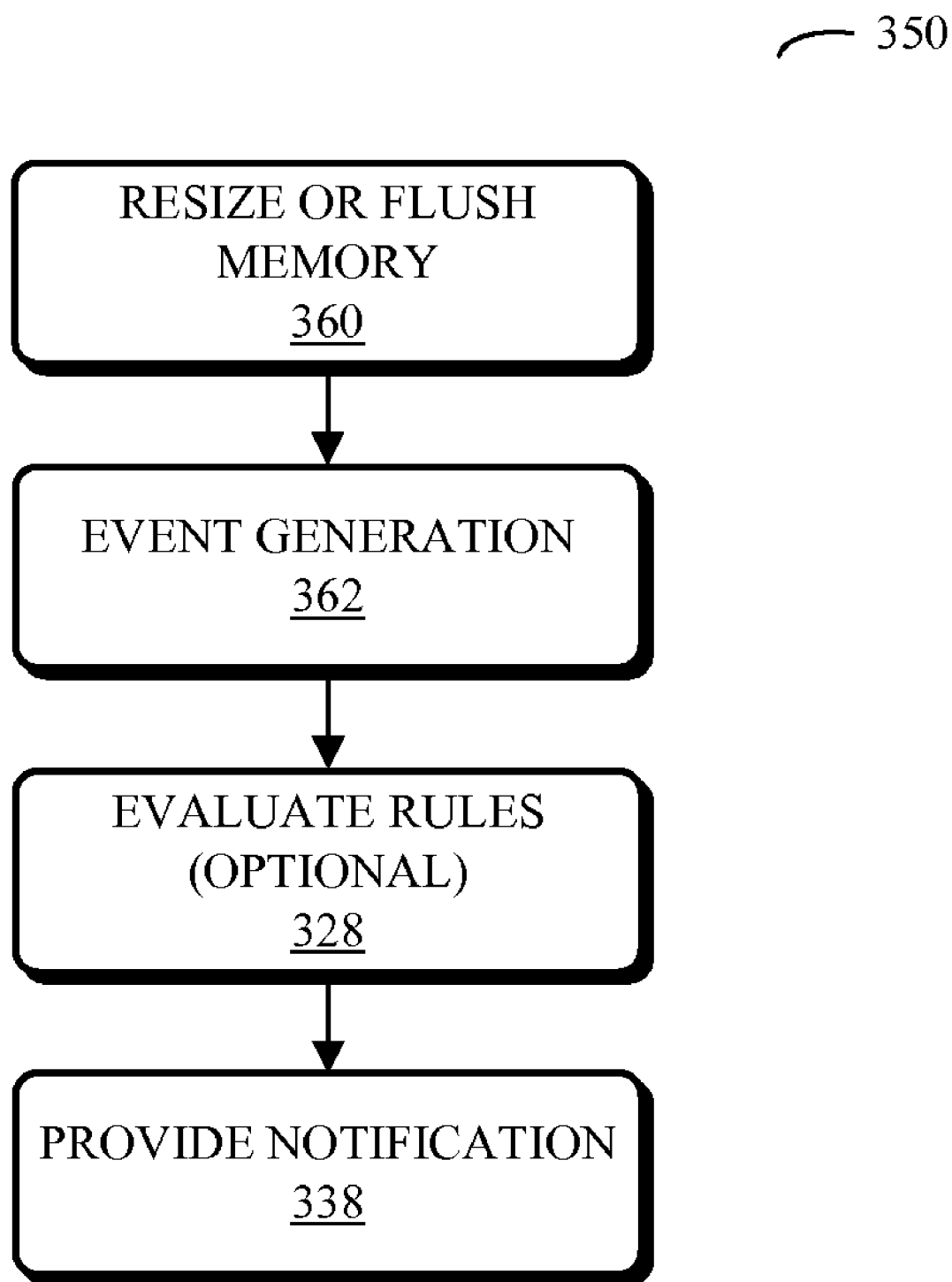


FIG. 3A

**FIG. 3B**

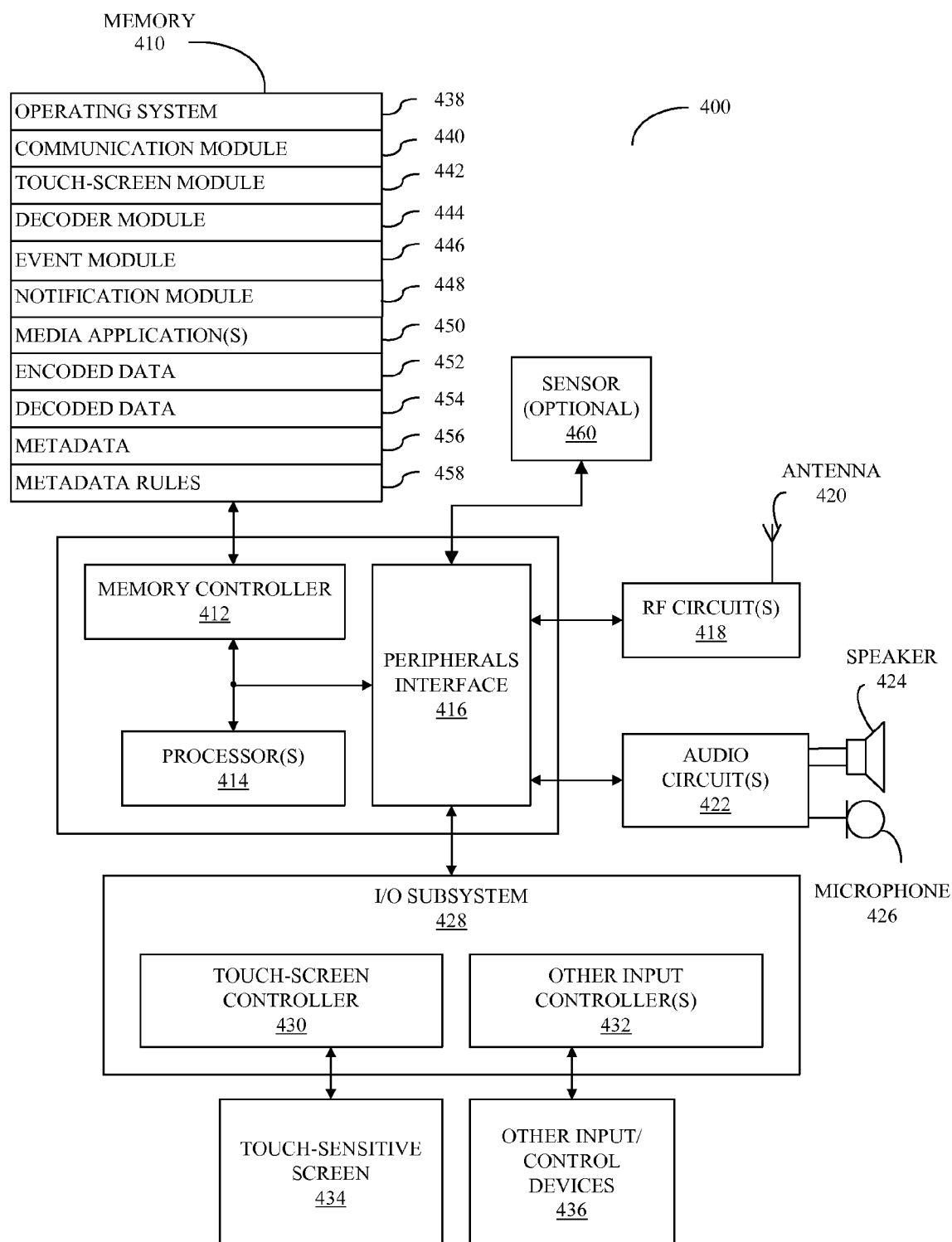


FIG. 4

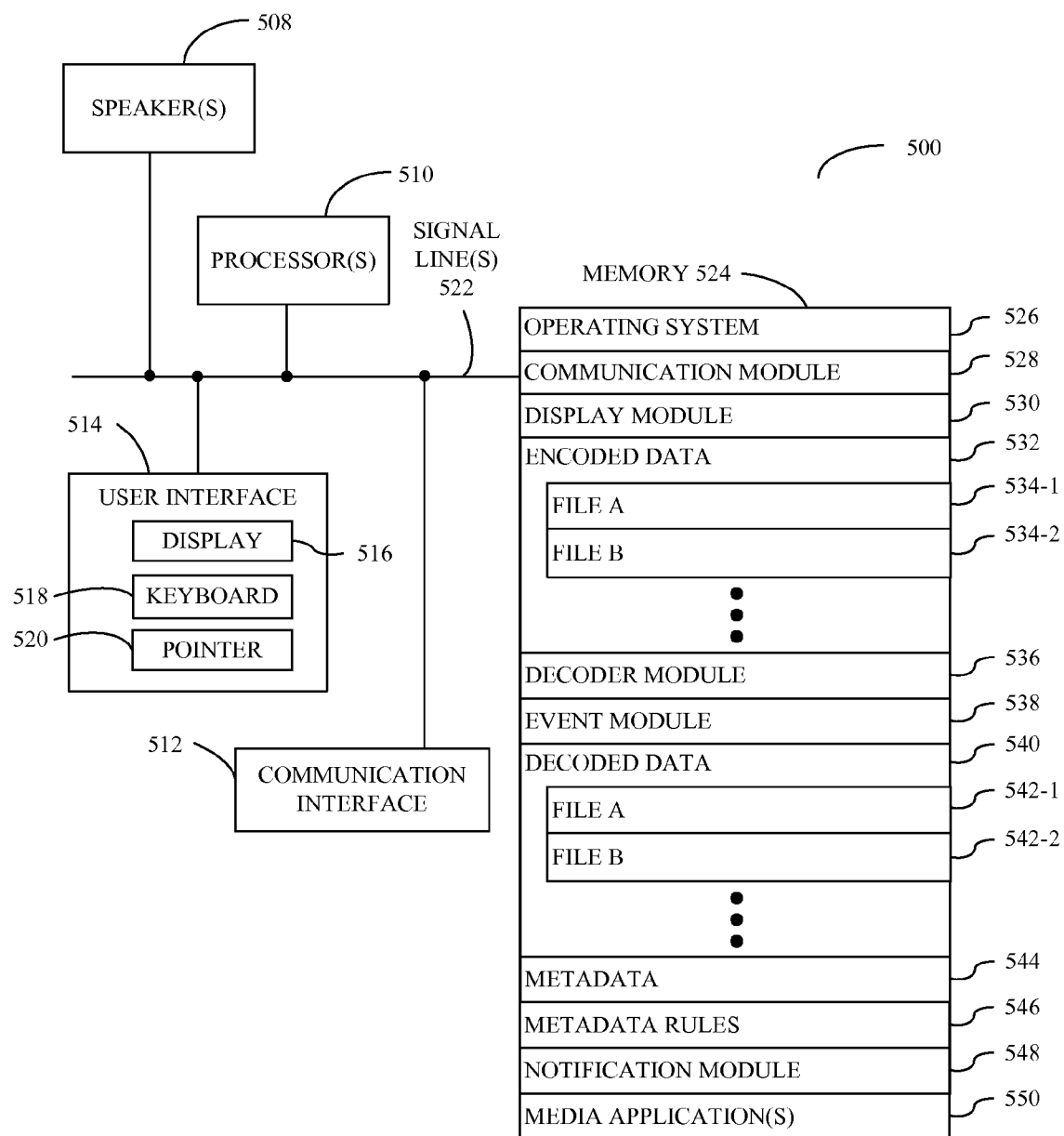


FIG. 5



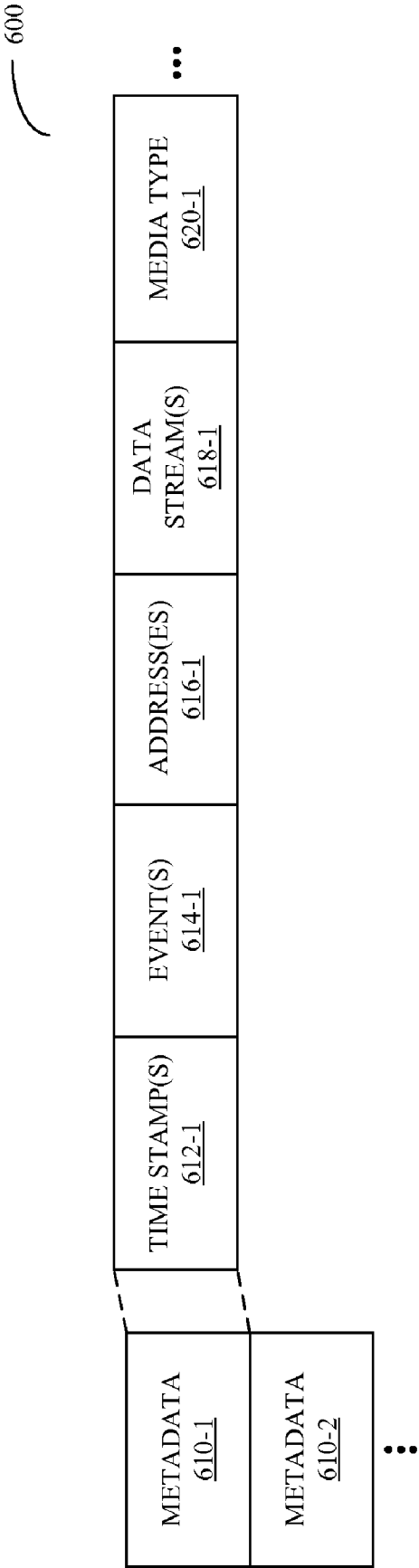


FIG. 6

## USING TIME-STAMPED EVENT ENTRIES TO FACILITATE SYNCHRONIZING DATA STREAMS

### BACKGROUND

#### [0001] 1. Field of the Invention

[0002] The present invention relates to techniques for synchronizing data streams. More specifically, the present invention relates to circuits and methods for storing information associated with events while processing data streams and using the stored information to synchronize the data streams.

#### [0003] 2. Related Art

[0004] Rapid advances in computing power and network bandwidth have enabled computer systems to easily manipulate various types of encoded media, such as encoded video files and encoded audio files. The process of playing back of encoded media often involves decoding different types of media, such as video images and associated audio. However, these different types of media are typically encoded in different formats. Consequently, decoding of encoded media often involves coordinating the decoding of separate data streams associated with the different types of media.

[0005] The task of coordinating separate data streams as they are processed in different decoders can be a challenge, especially as operating conditions and user requirements vary dynamically. Existing techniques attempt to coordinate different data streams by estimating when events occur or will occur during processing of the data streams. However, these estimates are often inaccurate, which can cause delays in providing decoded data to the user or can cause a glitch during playback that degrades the quality of the user experience.

[0006] Hence what is needed is a method and an apparatus that facilitates synchronizing data streams without the above-described problems.

### SUMMARY

[0007] Embodiments of a system that synchronizes multiple data streams (such as a video data stream and an associated audio data stream) are described. In particular, the system may store metadata multiple times as these data streams are processed. For example, the metadata may be stored as a given data stream is processed in a data pipeline, such as a decoder. Note that the metadata may be associated with data in at least a portion of the given data stream and may include: timestamps, one or more memory addresses associated with this portion of the data, and/or one or more events (such as when the data is decoded). Moreover, control logic or software in the system may use at least some of the stored metadata to improve the synchronization, thereby reducing or eliminating delays and/or glitches associated with timing mismatches between the data streams.

[0008] Another embodiment of the present invention provides the system that synchronizes data streams. During operation, the system receives a first data stream having a first data type and performs a sequence of operations on at least a portion of the first data stream. Next, the system stores metadata associated with the first data stream one or more times during the sequence of operations. Note that a given metadata entry includes a time stamp, data-stream information associated with the portion of the first data stream, and event information associated with the occurrence of one or more events during the sequence of operations. Moreover, the system provides a notification based on the stored metadata, the

notification to facilitate synchronizing the first data stream with a second data stream having a second data type.

[0009] In some embodiments, the first data stream includes initial metadata. Moreover, the first data stream may include a first type of media and the second data stream may include a second type of media. For example, the first data stream may include audio and the second data stream may include video corresponding to the audio.

[0010] In some embodiments, the sequence of operations includes decoding of audio data. Moreover, in some embodiments, the stored metadata indicates where the portion of the first data stream is located with reference to the sequence of operations.

[0011] In some embodiments, the system phase shifts the notification by a time interval, where the phase shift can be positive or negative.

[0012] In some embodiments, the data-stream information associated with the portion of the first data stream includes an address in a memory.

[0013] In some embodiments, the system evaluates multiple events in the stored metadata using pre-defined rules and generates the notification.

[0014] In some embodiments, the notification is provided based on a resizing of a memory allocation associated with at least one operation in the sequence of operations. Moreover, in some embodiments the notification is provided based on a presence of a given event in a subset of the one or more events in the stored metadata.

[0015] In some embodiments, the one or more events include: when the portion of the first data is consumed at an output at the end of the sequence of operations; completion of a given operation in the sequence of operations; completion of the sequence of operations; detection of an end of the first data stream; a time-scale modification affecting at least one operation in the sequence of operations; changing an active audio-output device; an error associated with the first data or at least one operation in the sequence of operations; an internal event associated with at least one operation in the sequence of operations or an external event associated with an application that consumes processed first data output from the sequence of operations; and/or a change in current media playback time. For example, the change may include: fast forwarding, rewinding and/or seeking.

[0016] In another embodiment of the system, the system provides the first data stream and the second data stream, and executes at least one event with respect to the first data stream. Next, the system provides a metadata entry for each event and uses the metadata entry to synchronize the first data stream with the second data stream.

[0017] Another embodiment provides a computer program product for use in conjunction with the system. This computer program product may include instructions corresponding to at least some of the above-described operations. Moreover, these instructions may include high-level code in a program module and/or low-level code that is executed by a processor in the system.

[0018] Another embodiment provides a method for synchronizing data streams. This method may perform at least some of the above-described operations.

[0019] Another embodiment provides one or more integrated circuits, which may be included in the system. During operation of the one or more integrated circuits, an input interface receives the first data stream having the first data type. Next, an operation circuit, which is electrically coupled

to the input interface, receives at least a portion of the first data stream and performs the sequence of operations on at least the portion of the first data stream. Then, an event circuit, which is electrically coupled to the operation circuit and a memory buffer, stores the metadata associated with at least the portion of the first data stream in the memory buffer one or more times during the sequence of operations. Moreover, a notification circuit (which is electrically coupled to the memory buffer) provides the notification based on the stored metadata, and an output interface (which is electrically coupled to the operation circuit and the notification circuit) provides processed data output by the operation circuit and the notification.

**[0020]** Another embodiment provides another integrated circuit, which may be included in the system. During operation of this integrated circuit, control logic stores the metadata associated with the first data stream in the memory while the sequence of operations is performed on at least a portion of the first data stream. Note that the metadata is stored one or more times during the sequence of operations, and the metadata includes temporal event information associated with the occurrence of one or more events during the sequence of operations which facilitates synchronizing the first data stream with the second data stream.

**[0021]** Another embodiment provides a portable device. This portable device may include one or more of the above-described integrated circuits and/or may execute instructions corresponding to at least some of the above-described operations. In some embodiments, the portable device includes a touch-sensitive display which is configured to determine movement of one or more points of contact by a user of the touch-sensitive display.

#### BRIEF DESCRIPTION OF THE FIGURES

**[0022]** FIG. 1 is a block diagram illustrating a circuit in accordance with an embodiment of the present invention.

**[0023]** FIG. 2A is a flowchart illustrating a process for synchronizing data streams in accordance with an embodiment of the present invention.

**[0024]** FIG. 2B is a flowchart illustrating a process for synchronizing data streams in accordance with an embodiment of the present invention.

**[0025]** FIG. 3A is a flowchart illustrating a process for synchronizing data streams in accordance with an embodiment of the present invention.

**[0026]** FIG. 3B is a flowchart illustrating a process for synchronizing data streams in accordance with an embodiment of the present invention.

**[0027]** FIG. 4 is a block diagram illustrating a portable device in accordance with an embodiment of the present invention.

**[0028]** FIG. 5 is a block diagram illustrating a computer system in accordance with an embodiment of the present invention.

**[0029]** FIG. 6 is a block diagram illustrating a data structure in accordance with an embodiment of the present invention.

**[0030]** Note that like reference numerals refer to corresponding parts throughout the drawings.

#### DETAILED DESCRIPTION

**[0031]** The following description is presented to enable any person skilled in the art to make and use the invention, and is provided in the context of a particular application and its requirements. Various modifications to the disclosed embodi-

ments will be readily apparent to those skilled in the art, and the general principles defined herein may be applied to other embodiments and applications without departing from the spirit and scope of the present invention. Thus, the present invention is not intended to be limited to the embodiments shown, but is to be accorded the widest scope consistent with the principles and features disclosed herein.

**[0032]** Embodiments of hardware, software, and/or processes for using the hardware and/or software are described. Note that hardware may include a circuit, a device (such as a portable device), and/or a system (such as a computer system), and software may include a computer program product for use with the computer system. Moreover, in some embodiments the portable device and/or the system include one or more of the circuits (for example, in one or more integrated circuits).

**[0033]** These circuits, devices, systems, computer program products, and/or processes may be used to synchronize multiple data streams (such as a video data stream and an associated audio data stream). In particular, metadata may be stored multiple times as these data streams are processed. For example, the metadata may be stored as a given data stream is processed in a data pipeline, such as a decoder. Note that the metadata may be associated with data in at least a portion of the given data stream and may include: timestamps, one or more memory addresses associated with this portion of the data, and/or one or more events (such as when the data is decoded).

**[0034]** Moreover, control logic or software in a device and/or a system may use at least some of the stored metadata to improve the synchronization, thereby reducing or eliminating delays and/or glitches associated with timing mismatches between the data streams. For example, if a video data is decoded before an associated audio data stream, the device and/or system may flash a displayed video image while waiting for the audio data stream. By reducing or eliminating the delays and/or glitches, these techniques may improve the quality of a user experience when using the device and/or system.

**[0035]** These techniques may be used in a wide variety of devices and/or systems. For example, the device and/or the system may include: a personal computer, a laptop computer, a cellular telephone, a personal digital assistant, an MP3 player, a portable television, an iPod (a trademark of Apple, Inc.), an iPhone, and/or a device that plays back one or more types of media.

**[0036]** Circuits that synchronize data streams in a device and/or a system in accordance with embodiments of the invention are now described. In the embodiments that follow, the technique is used to synchronize a video data stream with an associated audio data stream (which includes audio data, such as one or more audio files) as the video data stream is processed in a video decoder pipeline and the audio data stream is processed in an audio decoder pipeline. In other embodiments the technique is applied to synchronize data streams including different types of media which are processed in another decoder pipeline. In general, these techniques may be applied to an arbitrary type of data in a given data stream and/or during an arbitrary data pipeline or sequence of operations. For example, the techniques may be used to synchronize data streams associated with encryption and/or decryption of data. Note that the one or more circuits

may be included on one or more integrated circuits, and that the one or more integrated circuits may be included in the device and/or the system.

**[0037]** FIG. 1 presents a block diagram illustrating an embodiment of a circuit 100. Decoder 114 in this circuit may receive encoded data from memory 110 (such as a buffer) and may provide decoded data to memory 116. In an exemplary embodiment, the encoded data includes encoded audio data, which may be associated with a file (e.g., a song or an album). This audio data may be compatible with a variety of encoding or file formats, including: Advance Audio Coding (AAC), High Efficiency Advance Audio Coding (HE-AAC), an MPEG standard (such as MP3), Algebraic Code Excited Linear Prediction (ACELP), Apple Lossless Audio Codec (ALAC), Wave (WAV), Audio Interchange File Format (AIFF), Adaptive Multi-Rate (AMR), an Interactive Media Association (IMA) standard, and/or a QDesign Music Codec, as well as other encoding or file formats. However, note that the circuit 100 may be used to decode a variety of types of media, such as video and/or encrypted data.

**[0038]** Decoder 114 may use a portion of memory 110 (such as memory 112) during the receiving of the encoded data and/or a portion of memory 116 (such as memory 118) during the decoding of the encoded data. For example, memory 112 and/or memory 118 may be used as temporary memory during the decoding. In some embodiments, memory 112 includes a different type of memory than memory 110, and memory 118 includes a different type of memory than memory 116. For example, memory 112 and/or 118 maybe SRAM and memory 110 and/or 116 may be DRAM. However, a wide variety of types of memory may be used for these components in the circuit 100, including: DRAM, SRAM, FLASH, solid-state memory, volatile memory, and/or non-volatile memory. Moreover, some or all of memory 110, 112, 116, and/or 118 may be separate components or integrated with one another into a single component.

**[0039]** Memory 116 may store the decoded data until it is consumed by hardware consumer 120 (such as one or more audio circuits and speakers) on behalf of a media playback application or software that executes in the device and/or the system which includes the circuit 100. When the decoded data is consumed (e.g., the decoded data is output to the hardware consumer 120), the consumed decoded data may be removed from the memory 116. Alternatively, consumed decoded data is no longer needed in the memory 116 and may be subsequently overwritten or erased.

**[0040]** In general, a rate of processing the encoded data in the circuit 100 may vary based on user commands and dynamic operation of the device and/or the system. For example, a user may instruct a playback application or software to perform a seek operation, which may move a pointer within a given file or to another file. Moreover, the user may perform a time-scale modification operation when post processing decoded data. Consequently, the decoder 114 may be reset as the encoded data to be decoded is changed. However, these dynamic modifications make it difficult to estimate or predict when encoded data associated with the new pointer location is processed and output to the memory 116. Thus, during operation of the device and/or system it may be difficult to coordinate or synchronize a data stream that includes decoded data output to the hardware consumer 120 with

another data stream that includes another type of decoded data (such as video data) associated with another data pipeline.

**[0041]** One approach to reduce or eliminate synchronization errors associated with the different data streams is to store metadata. In particular, as a given data stream is processed in the circuit 100, metadata associated with at least a portion of the data in the given data stream may be stored when different events occur. This metadata may be used by control logic 122 to synchronize the data stream with one or more additional data streams.

**[0042]** As illustrated in FIG. 1, when an event occurs, event generator 128 may add metadata (including one or more event tokens) to the decoded data (or the data being decoded). This metadata may be stored in memory 116 and/or 118. In general, such metadata may be stored one or more times as a sequence of operations is performed on a data stream traversing the circuit 100. Control logic 122 may use this metadata to provide one or more notifications (such as synchronization commands) to an application 124 (such as a media playback application) and/or a client 126 which is associated with consumption of the decoded data by the hardware consumer 120. For example, the control logic 122 may initiate or queue additional operations based on the stored metadata, the time stamps, and/or when the decoded data in the memory 116 is consumed. Alternatively, a video signal may be resynchronized based on the one or more notifications, thereby preventing video from being displayed prematurely (relative to an associated audio data stream). Consequently, these notifications may facilitate improved synchronization of an output data stream of decoded data from the memory 116 (such as decoded audio data) with another data stream (such as decoded video data).

**[0043]** In some embodiments, a given metadata entry includes a time stamp, data-stream information associated with the portion of the data stream (such as one or more memory addresses where data associated with operations in the sequence of operations is stored), and event information associated with the occurrence of one or more events during the sequence of operations. Moreover, in some embodiments the stored metadata indicates where the portion of the data stream is in the sequence of operations. In general, the metadata may include temporal event information associated with the occurrence of one or more events during the sequence of operations which facilitates synchronizing the data stream with another data stream.

**[0044]** A wide variety of events during the decoding (or more generally, the sequence of operations) may result in metadata being stored. For example, the one or more events may include: completion of a given operation during the decoding, completion of the decoding of encoded data, and/or when decoded data is consumed from the memory 116. Moreover, an event may be associated with a time-scale modification affecting at least one operation during the decoding. This time-scale modification may change a current media playback time, such as a trick mode (for example, fast forward, rewind, seeking, 2x, and/or pause).

**[0045]** In some embodiments, changing an active audio-output device associated with the hardware consumer 120 constitutes an event and results in metadata being stored. For example, such a device change may include: stopping audio playback, switching the device, restarting the audio playback, and notifying the device and/or the system of the change.

[0046] Other device changes may include resizing of a display and/or the occurrence of an error. For example, an error associated with an operation during the decoding may constitute an event and may result in metadata being stored. In general, the events (and thus the metadata) may include one or more internal events associated with at least one operation during the decoding or one or more external events associated with the hardware consumer 120, the application 124, and/or the client 126, which consume decoded data from the memory 116.

[0047] As described below with reference to FIGS. 3A and 3B, control logic 122 may evaluate multiple events in the stored metadata. In some embodiments, this analysis is based on pre-defined rules, which prioritize the different events and determine the appropriate notification(s) to generate. For example, some events, such as a time-scale modification or a resizing of an amount of memory allocated to the decoded data, may have a greater impact on synchronizing different data streams in the device and/or the system. Consequently, if one of these events is present in the metadata, the notification(s) may be based on these events. However, in other embodiments control logic 122 may interpolate between the time stamps in the metadata to determine how to synchronize the different data streams using the one or more notifications.

[0048] In this way, the metadata allows the control logic 122 to determine latency associated with operation of the circuit 100 (and, more generally, with the sequence of operations associated with the decoding). Moreover, this synchronization technique is able to account for, and thus, to include the impact of generalized events which occur internally or externally to the circuit 100 but which impact the latency associated with the decoding. Consequently, by storing metadata associated with the one or more events and subsequently generating notifications or synchronization commands based on one or more of the events in the stored metadata, the circuit 100 may be used to reduce or eliminate synchronization errors between different data streams.

[0049] Note that in some embodiments the circuit 100 includes fewer or additional components. Moreover, two or more components can be combined into a single component and/or a position of one or more components can be changed. In some embodiments, some or all of the functions illustrated in the circuit 100 are implemented in software.

[0050] Processes for synchronizing data streams, which may be performed by a device and/or a system, in accordance with embodiments of the invention are now described. FIG. 2A presents a flowchart illustrating an embodiment of a process 200 for synchronizing data streams, which may be performed by the device and/or the system. During operation, the system receives a first data stream having a first data type (210) and performs a sequence of operations on at least a portion of the first data stream (212). Next, the system stores metadata associated with the first data stream one or more times during the sequence of operations (214). Note that a given metadata entry includes a time stamp, data-stream information associated with the portion of the first data stream, and event information associated with the occurrence of one or more events during the sequence of operations.

[0051] In some embodiments, the system optionally evaluates multiple events in the stored metadata and generates a notification based on the evaluation (216). Moreover, the system may phase shift the notification by a time interval (218), where the phase shift is positive or negative. Then, the system provides the notification based on the stored metadata

(220), where the notification facilitates synchronizing the first data stream with a second data stream having a second data type, where the notification facilitates synchronizing the first data stream with a second data stream having a second data type.

[0052] In an exemplary embodiment, the first data stream includes encoded audio data and the sequence of operations includes decoding of this data. As the encoded data is decoded, metadata is stored. This metadata may include information about events, such as completion of a portion of the decoding or a change to a playback speed (such as when a user seeks through an audio file). Moreover, this metadata may be analyzed to estimate when the decoding will be completed, and to generate one or more notifications that are used to synchronize a stream of decoded data with another stream, such as one that includes decoded video.

[0053] FIG. 2B presents a flowchart illustrating an embodiment of a process 230 for synchronizing data streams, which may be performed by the device and/or the system. During operation, the system provides a first data stream and a second data stream (240), and executes at least one event with respect to the first data stream (242). Next, the system provides a metadata entry for each event (244) and uses the metadata entry to synchronize the first data stream with the second data stream (246).

[0054] FIG. 3A presents a flowchart illustrating an embodiment of a process 300 for synchronizing data streams, which may be implemented by the device and/or the system. During operation, the system receives encoded data (308) and decodes the encoded data (310). Note that this encoded data may include initial metadata, such as a time stamp that is included in a file format and/or an event token (such as an end-of-file marker). Moreover, based on one or more inputs 334, the system may generate additional metadata such as event (332) token corresponding to an event (such as the occurrence of one of the inputs 334). In an exemplary embodiment, the one or more inputs 334 include: a time-scale modification (such as during post processing of decoded audio data), an effects change (such as equalization), a resume command (following a previous pause command), and/or a time (which may indicate that a user interface is to perform an operation a pre-defined time prior to outputting the decoded data).

[0055] If the system is currently running (336) or performing a sequence of operations (for example, the system is currently consuming decoded data generated during the process 300), the additional metadata may be added to the decoded data. As discussed further below, if the system is not currently running (336) (for example, the user has paused consumption of decoded data), the system may store the additional metadata in a stored list of metadata (not shown) and/or may provide one or more notifications (338) based on the stored metadata. This notification(s) may be used to synchronize different data streams.

[0056] Next, the system may post process the decoded data (312). This post processing may include equalization and/or certain trick modes (such as jumps). Then, the system may determine if one or more event tokens associated with the initial metadata and/or additional metadata are included in the decoded data stream (314). If yes, the initial metadata and/or the additional metadata are added to the stored list (316) of metadata associated with at least a portion of the decoded data. However, whether or not the one or more event tokens associated with the initial metadata and/or the additional

metadata are detected (314), the decoded data is written to a buffer (318). Note that the time stamp(s) in the initial and/or additional metadata may be updated as the data progresses through the process 300, when the metadata in the stored list is updated (316), and/or when decoded data is consumed. Moreover, note that additional entries to the stored metadata may be appended to the end of the stored list and/or may be interleaved between existing entries in the stored metadata.

[0057] Subsequently, the decoded data may be consumed from the buffer (320). Based on one or more inputs 326 (which may occur prior, during, or after the decoded data is consumed), the system may generate additional metadata such as another event (324) token. For example, the one or more inputs 326 may include: an end of the decoding, changing the device that is consuming the decoded data, and/or an error (such as a decoder error or a read error while filling encoded data into the memory 110 in FIG. 1).

[0058] Next, if the one or more event tokens associated with the additional metadata are detected (322), the additional metadata may be added to the stored list (not shown) of metadata associated with at least a portion of the decoded data. Moreover, the various stored event tokens may be optionally evaluated (328) based on one or more pre-defined rules. For example, these rules may prioritize the various events to determine a type of one or more notification to provide (338). Note that in some embodiments the one or more notifications may be time shifted earlier or later by optionally modifying timing (330). This time shifting may allow interpolation of the timing of the one or more notifications, such as when the decoded data is being rapidly consumed.

[0059] Modifications to the memory 110 (FIG. 1), 112 (FIG. 1), 116 (FIG. 1), and/or 118 (FIG. 1) may also constitute an event. For example, an amount of memory allocated to decoded data may be changed based on a sampling rate or a sample size of the data, or decoded data in memory may be flushed during a seek operation. However, such events may be handled in a separate (but related) process from the process 300. This is shown in FIG. 3B, which presents a flowchart illustrating an embodiment of a process 350 for synchronizing data streams, which may be implemented by the device and/or the system.

[0060] During operation, the system may detect a resizing or flushing of data (360) from a memory associated with the circuit 100 (FIG. 1) and additional metadata (including an event token) may be generated (362) and added to the stored list of metadata (not shown). Next, the system may optionally evaluate one or more of the stored event tokens based on the one or more pre-defined rules (328) (for example, a latency may be determined following the changing the allocated memory) and may provide the notification (338).

[0061] Note that in some embodiments of the processes 200 (FIG. 2A), 230 (FIG. 2B), 300 (FIG. 3A), and/or 350 there may be additional or fewer operations, the order of the operations may be changed, and two or more operations may be combined into a single operation.

[0062] Devices and computer systems for implementing these synchronization techniques in accordance with embodiments of the invention are now described. FIG. 4 presents a block diagram illustrating an embodiment of a portable device 400, which may include a touch-sensitive screen 434. This device may include a memory controller 412, one or more data processors, image processors and/or central processing units 414, and a peripherals interface 416. Moreover,

the memory controller 412, the one or more processors 414, and/or the peripherals interface 416 may be separate components or may be integrated, such as on one or more integrated circuits. Note that the various components in the portable device 400 may be electrically coupled by one or more signal lines and/or communication buses.

[0063] Peripherals interface 416 may be electrically coupled to: an optional sensor 460 (such as CMOS or CCD image sensor), one or more RF circuits 418, one or more audio circuits 422, and/or an input/output (I/O) subsystem 428. These audio circuits 422 may be electrically coupled to a speaker 424 and a microphone 426. Note that the portable device 400 may support voice recognition and/or voice replication.

[0064] Moreover, the RF circuits 418 may be electrically coupled to one or more antennas 420 and may allow communication with one or more additional devices, computers and/or servers using a wireless network. Consequently, in some embodiments portable device 400 supports one or more communication protocols, including: code division multiple access (CDMA), global system for mobile communications (GSM), Enhanced Data GSM Environment (EDGE), Wi-Fi (such as IEEE 802.11a, IEEE 802.11b, IEEE 802.11g, and/or IEEE 802.11n), Bluetooth, Wi-MAX, a protocol for email, instant messaging, a simple message system (SMS), and/or any other suitable communication protocol (including communication protocols not yet developed as of the filing date of this document). In an exemplary embodiment, the portable device 400 is, at least in part, a cellular telephone.

[0065] In some embodiments, I/O subsystem 428 includes a touch-screen controller 430 and/or other input controller(s) 432. This touch-screen controller may be electrically coupled to a touch-sensitive screen 434. Moreover, the touch-sensitive screen 434 and the touch-screen controller 430 may detect contact and any movement or break thereof using any of a plurality of touch-sensitivity technologies, including but not limited to: capacitive, resistive, infrared, and/or surface acoustic wave technologies, as well as other proximity sensor arrays or other elements for determining one or more points of contact with the touch-sensitive screen 434. In an exemplary embodiment, the touch-sensitive screen 434 has a resolution in excess of 100 dpi, such as approximately 168 dpi.

[0066] Note that the other input controller(s) 432 may be electrically coupled to other input/control devices 436, such as: one or more physical buttons, a keyboard, an infrared port, a USB port, and/or a pointer device (such as a mouse). Moreover, the one or more physical buttons may include an up/down button for volume control of the speaker 424 and/or the microphone 426.

[0067] In some embodiments, the one or more physical buttons include a push button. By quickly pressing the push button, a user of the portable device 400 may disengage locking of the touch-sensitive screen 434. Alternatively, by pressing the push button for a longer time interval, the user may turn power to the portable device 400 on or off. Moreover, the touch-sensitive screen 434 may be used to implement virtual or soft buttons and/or a keyboard. Note that the user may be able to customize a functionality of one or more of the virtual and/or physical buttons.

[0068] In some embodiments, the portable device 400 includes circuits for supporting a location determining capability, such as that provided by the global positioning system (GPS). Moreover, the portable device 400 may be used to play back recorded music, such as one or more files, including

MP3 files or AAC files. Consequently, in some embodiments the portable device 400 includes the functionality of an MP3 player, such as an iPod (trademark of Apple, Inc.). Therefore, the portable device 400 may include a connector that is compatible with the iPod™.

[0069] Memory controller 412 may be electrically coupled to memory 410. Memory 410 may include high-speed random access memory and/or non-volatile memory, such as: one or more magnetic disk storage devices, one or more optical storage devices, and/or FLASH memory. Memory 410 may store an operating system 438, such as: Darwin, RTXC, LINUX, UNIX, OS X, Windows, and/or an embedded operating system such as VxWorks. This operating system may include procedures (or sets of instructions) for handling basic system services and for performing hardware-dependent tasks. Moreover, memory 410 may also store communication procedures (or sets of instructions) in a communication module 440. These communication procedures may be used for communicating with one or more additional devices, one or more computers and/or one or more servers.

[0070] Memory 410 may include a touch-screen module 442 (or a set of instructions), a decoder module 444 (or a set of instructions), event module 446 (or a set of instructions), and/or a notification module 448 (or a set of instructions). Touch-screen module 442 may provide graphics associated with the virtual buttons and/or keyboard. Moreover, the decoder module 444 may receive encoded data 452 to produce decoded data 454, which is consumed by one or more media applications 450.

[0071] In some embodiments, event module 446 may detect and/or generate one or more event tokens (each of which corresponds to an event). These event tokens may be stored with metadata 456 associated with the encoded data 452 and/or the decoded data 454. Moreover, notification module 448 may generate one or more notifications by evaluating the stored event tokens in the metadata 456 based on metadata rules 458. This notification(s) may be used to synchronize different data streams, such as a data stream of decoded audio data sent to speaker 424 and a data stream of decoded video data sent to touch-sensitive screen 434.

[0072] Note that each of the above-identified modules and applications corresponds to a set of instructions for performing one or more functions described above. These modules (e.g., sets of instructions) need not be implemented as separate software programs, procedures or modules. Consequently, the various modules and sub-modules may be rearranged and/or combined. Moreover, memory 410 may include additional modules and/or sub-modules, or fewer modules and/or sub-modules. Therefore, memory 410 may include a subset or a superset of the above-identified modules and/or sub-modules.

[0073] Moreover, instructions in the various modules in the memory 410 may be implemented in a high-level procedural language, an object-oriented programming language, and/or in an assembly or machine language. The programming language may be compiled or interpreted, e.g., configurable or configured to be executed by the one or more processing units 414. Consequently, the instructions may include high-level code in a program module and/or low-level code, which is executed by the processor(s) 414 in the portable device 400. Note that various functions of the device 400 may be implemented in hardware and/or in software, including in one or more signal processing and/or application-specific integrated circuits.

[0074] FIG. 5 presents a block diagram illustrating an embodiment of a computer system 500. Computer system 500 can include: one or more processors 510, a communication interface 512, a user interface 514, speakers 508, and/or one or more signal lines 522 electrically coupling these components together. Note that the one or more processing units 510 may support parallel processing and/or multi-threaded operation, the communication interface 512 may have a persistent communication connection, and the one or more signal lines 522 may constitute a communication bus. Moreover, the user interface 514 may include: a display 516, a keyboard 518, and/or a pointer 520, such as a mouse.

[0075] Memory 524 in the computer system 500 may include volatile memory and/or non-volatile memory. More specifically, memory 524 may include: ROM, RAM, EPROM, EEPROM, FLASH, one or more smart cards, one or more magnetic disc storage devices, and/or one or more optical storage devices. Memory 524 may store an operating system 526 that includes procedures (or a set of instructions) for handling various basic system services for performing hardware-dependent tasks. Memory 524 may also store communication procedures (or a set of instructions) in a communication module 528. These communication procedures may be used for communicating with one or more computers and/or servers, including computers and/or servers that are remotely located with respect to the computer system 500.

[0076] Memory 524 may include multiple program modules (or a set of instructions), including: display module 530 (or a set of instructions), decoder module 536 (or a set of instructions), event module 538 (or a set of instructions), and/or notification module 548 (or a set of instructions). Display module 530 may provide graphics for display on display 516. Moreover, the decoder module 536 may receive encoded data 532 (such as file A 534-1 and/or file B 534-2) and may produce decoded data 540 (such as file A 542-1 and/or file B 542-2), which is consumed by one or more media applications 550.

[0077] In some embodiments, event module 538 may detect and/or generate one or more event tokens (each of which corresponds to an event). These event tokens may be stored with metadata 544 associated with the encoded data 532 and/or the decoded data 540. Moreover, notification module 548 may generate one or more notifications by evaluating the stored event tokens in the metadata 544 based on metadata rules 546. This notification(s) may be used to synchronize different data streams, such as a data stream of decoded audio data sent to speakers 508 and a data stream of decoded video data sent to display 516.

[0078] Instructions in the various modules in the memory 524 may be implemented in a high-level procedural language, an object-oriented programming language, and/or in an assembly or machine language. The programming language may be compiled or interpreted, e.g., configurable or configured to be executed by the one or more processing units 510. Consequently, the instructions may include high-level code in a program module and/or low-level code, which is executed by the processor 510 in the computer system 500.

[0079] Although the computer system 500 is illustrated as having a number of discrete components, FIG. 5 is intended to provide a functional description of the various features that may be present in the computer system 500 rather than a structural schematic of the embodiments described herein. In practice, and as recognized by those of ordinary skill in the art, the functions of the computer system 500 may be distrib-

uted over a large number of servers or computers, with various groups of the servers or computers performing particular subsets of the functions. In some embodiments, some or all of the functionality of the computer system 500 may be implemented in one or more application-specific integrated circuits (ASICs) and/or one or more digital signal processors (DSPs). [0080] Computer system 500 may include fewer components or additional components. Moreover, two or more components can be combined into a single component and/or a position of one or more components can be changed. In some embodiments the functionality of the computer system 500 may be implemented more in hardware and less in software, or less in hardware and more in software, as is known in the art.

[0081] Data structures that may be used in the portable device 400 (FIG. 4) and/or the computer system 500 in accordance with embodiments of the invention are now described. FIG. 6 presents a block diagram illustrating an embodiment of a data structure 600. This data structure may include metadata 610, which may be used to synchronize different data streams. A given instance of the metadata, such as metadata 610-1, may include: one or more time stamps 612-1, one or more events 614-1, one or more addresses 616-1 (such as a memory address), portions of one or more data streams 618-1 with which the metadata is associated, and/or a media type 620-1.

[0082] Note that in some embodiments of the data structure 600 there may be fewer or additional components. Moreover, two or more components can be combined into a single component and/or a position of one or more components can be changed.

[0083] The foregoing descriptions of embodiments of the present invention have been presented for purposes of illustration and description only. They are not intended to be exhaustive or to limit the present invention to the forms disclosed. Accordingly, many modifications and variations will be apparent to practitioners skilled in the art. Additionally, the above disclosure is not intended to limit the present invention. The scope of the present invention is defined by the appended claims.

What is claimed is:

1. A method for synchronizing data streams, comprising: receiving a first data stream having a first data type; performing a sequence of operations on at least a portion of the first data stream; storing metadata associated with the first data stream one or more times during the sequence of operations, a given metadata entry including a time stamp, data-stream information associated with the portion of the first data stream, and event information associated with the occurrence of one or more events during the sequence of operations; and providing a notification based on the stored metadata to facilitate synchronizing the first data stream with a second data stream having a second data type.
2. The method of claim 1, wherein the first data stream includes initial metadata.
3. The method of claim 1, wherein the first data stream includes audio and the second data stream includes video corresponding to the audio.
4. The method of claim 1, wherein the first data stream includes data associated with a first type of media and the second data stream includes data associated with a second type of media.

5. The method of claim 1, wherein the sequence of operations includes decoding audio data.

6. The method of claim 1, wherein the stored metadata indicates where the portion of the first data stream is located with reference to the sequence of operations.

7. The method of claim 1, further comprising phase shifting the notification by a time interval, the phase shift being positive or negative.

8. The method of claim 1, wherein the data-stream information associated with at least the portion of the first data stream includes an address in a memory.

9. The method of claim 1, further comprising: evaluating multiple events in the stored metadata using pre-defined rules; and generating the notification based on the evaluation.

10. The method of claim 1, wherein the notification is provided based on a resizing of a memory allocation associated with at least one operation in the sequence of operations.

11. The method of claim 1, wherein the one or more events includes when the portion of the first data is consumed at an output at the end of the sequence of operations.

12. The method of claim 1, wherein the one or more events includes completion of a given operation in the sequence of operations.

13. The method of claim 1, wherein the one or more events includes detection of an end of the first data stream.

14. The method of claim 1, wherein the one or more events includes a time-scale modification affecting at least one operation in the sequence of operations.

15. The method of claim 1, wherein the one or more events includes a change in current media playback time.

16. The method of claim 15, wherein the change includes fast forwarding, rewinding or seeking.

17. The method of claim 1, wherein the one or more events includes changing an active audio-output device.

18. The method of claim 1, wherein the one or more events includes completion of the sequence of operations.

19. The method of claim 1, wherein the one or more events includes an error associated with the first data or at least one operation in the sequence of operations.

20. The method of claim 1, wherein the one or more events include an internal event associated with at least one operation in the sequence of operations or an external event associated with an application that consumes processed first data output from the sequence of operations.

21. The method of claim 1, wherein the notification is provided based on a presence of a given event in a subset of the one or more events in the stored metadata.

22. A computer program product for use in conjunction with a computer system, the computer program product comprising a computer-readable storage medium and a computer-program mechanism embedded therein for synchronizing data streams, the computer-program mechanism including:

instructions for receiving a first data stream having a first data type;

instructions for performing a sequence of operations on at least a portion of the first data stream;

instructions for storing metadata associated with the portion of the first data stream one or more times during the sequence of operations; and

instructions for providing a notification based on the stored metadata, the notification to facilitate synchronizing the first data stream with a second data stream having a second data type.



23. A computer system to synchronize data streams, comprising:

- a processor;
- memory; and

- a program module, wherein the program module is stored in the memory and configurable to be executed by the processor, the program module including:

- instructions for receiving a first data stream having a first data type;

- instructions for performing a sequence of operations on at least a portion of the first data stream;

- instructions for storing metadata associated with the portion of the first data stream one or more times during the sequence of operations, a given metadata entry including a time stamp, data-stream information associated with the portion of the first data stream, and event information associated with the occurrence of one or more events during the sequence of operations; and

- instructions for providing a notification based on the stored metadata, the notification to facilitate synchronizing the first data stream with a second data stream having a second data type.

24. An integrated circuit, comprising:

- an input interface configured to receive a first data stream having a first data type;

- an operation circuit electrically coupled to the input interface, the operation circuit configured to receive at least a portion of the first data stream and to perform a sequence of operations on at least the portion of the first data stream;

- an event circuit electrically coupled to the operation circuit and a memory buffer, the event circuit configured to store metadata associated with at least the portion of the first data stream in the memory buffer one or more times during the sequence of operations, a given metadata entry including a time stamp, data-stream information associated with the portion of the first data stream, and event information associated with the occurrence of one or more events during the sequence of operations;

- a notification circuit electrically coupled to the memory buffer, the notification circuit configured to provide a notification based on the stored metadata, the notification to facilitate synchronizing the first data stream with a second data stream having a second data type; and

- an output interface electrically coupled to the operation circuit and the notification circuit, the output interface

configured to provide processed data output by the operation circuit and the notification.

25. A method for synchronizing data streams, comprising: providing a first data stream and a second data stream;

- executing at least one event with respect to the first data stream;

- providing a metadata entry for each event; and

- using the metadata entry to synchronize the first data stream with the second data stream.

26. An integrated circuit, comprising control logic configured to store metadata associated with a first data stream in a memory while a sequence of operations is performed on at least a portion of the first data stream,

- wherein the metadata is to be stored one or more times during the sequence of operations,

- wherein the metadata includes temporal event information associated with the occurrence of one or more events during the sequence of operations which facilitates synchronizing the first data stream with a second data stream.

27. A system, comprising:

- a first memory configured to receive a first data stream having a first data type;

- an operation circuit electrically coupled to the first memory, the operation circuit configured to receive at least a portion of the first data stream and to perform a sequence of operations on at least the portion of the first data stream;

- a second memory configured to receive processed data output by the operation circuit;

- an event circuit electrically coupled to the operation circuit and the second memory, the event circuit configured to store metadata associated with at least the portion of the first data stream in the second memory one or more times during the sequence of operations, a given metadata entry including a time stamp, data-stream information associated with the portion of the first data stream, and event information associated with the occurrence of one or more events during the sequence of operations; and

- a notification circuit electrically coupled to the second memory, the notification circuit configured to provide a notification based on the stored metadata, the notification to facilitate synchronizing the first data stream with second data stream having a second data type.

\* \* \* \* \*